Lab 4 - Interrupt Service Routine

Friday, March 27, 2020 4:11 PM

To use Interrupts, we need three things:

1. An interrupt source
2. We must Enable the specific Interrupt
3. We need an Interrupt Service Routine (ISR)

Global Variables and the timer ISR

```
int one_sec_count = 0;
int four_sec_count = 0;
int raw_count = 0;
int update_flag = 0;

void __ISR(_TIMER_1_VECTOR, ipl2) Timer1Handler(void)
{
	update_flag = 1;                                        // signal to main to process update
	raw_count += 1;                                         // increment raw clock at every interrupt (1/2 sec)
	PORTToggleBits(IOPORT_G,BIT_12);                // toggle LED 1 (2Hz)
	if((raw_count % 8) == 0)
		PORTToggleBits(IOPORT_G,BIT_15);       // toggle LED 4 (.25Hz)

	mT1ClearIntFlag();                                      // clear the interrupt flag
}
```

Continuous Loop Mechanism in main( )

```
/* Perform the main application loop.    */
while (1)
{
        if(update_flag)  {
                if ((raw_count % 2) == 0)
                        one_sec_count += 1;                     // divide raw_count by two
                if ((raw_count % 8) == 0)
                        four_sec_count += 1;            // divide raw_count by eight

                 // Mask off least sig. 8 bits
                count_bits = (short int)(one_sec_count & 0xff);

                // write bits to output port
                mPORTEWrite(count_bits);

                update_flag = 0;                                // clear update flag
        }

        Nop();                                                          // if no update, do other work
}
```